Deep Occlusion-Aware Instance Segmentation with Overlapping BiLayers

Lei Ke[1] Yu-Wing Tai[2] Chi-Keung Tang[1]
[1]The Hong Kong University of Science and Technology [2]Kuaishou Technology
{lkeab,cktang}@cse.ust.hk, yuwing@gmail.com

Abstract

*Segmenting highly-overlapping objects is challenging, because typically no distinction is made between real object contours and occlusion boundaries. Unlike previous two-stage instance segmentation methods, we model image formation as composition of two overlapping layers, and propose **B**ilayer **C**onvolutional **Net**work (**BCNet**), where the top GCN layer detects the occluding objects (occluder) and the bottom GCN layer infers partially occluded instance (occludee). The explicit modeling of occlusion relationship with bilayer structure naturally decouples the boundaries of both the occluding and occluded instances, and considers the interaction between them during mask regression. We validate the efficacy of bilayer decoupling on both one-stage and two-stage object detectors with different backbones and network layer choices. Despite its simplicity, extensive experiments on COCO and KINS show that our occlusion-aware BCNet achieves large and consistent performance gain especially for heavy occlusion cases. Code is available at https://github.com/lkeab/BCNet.*

1. Introduction

State-of-the-art approaches in instance segmentation often follow the Mask R-CNN [21] paradigm with the first stage detecting bounding boxes, followed by the second stage to segment instance masks. Mask R-CNN and its variants [42, 5, 8, 25, 7] have demonstrated notable performance, and most of the leading approaches in the COCO instance segmentation challenge [40] have adopted this pipeline. However, we note that most incremental improvement comes from better backbone architecture designs, with little attention paid in the instance mask regression after obtaining the ROI (Region-of-Interest) features from object detection. We observe that a lot of segmentation errors are caused by overlapping objects, especially for object instances belonging to the same class. This is because each instance mask is individually regressed, and the regression process implicitly assumes the object in an ROI has almost complete contour, since most objects in the training data in COCO do not exhibit significant occlusions.

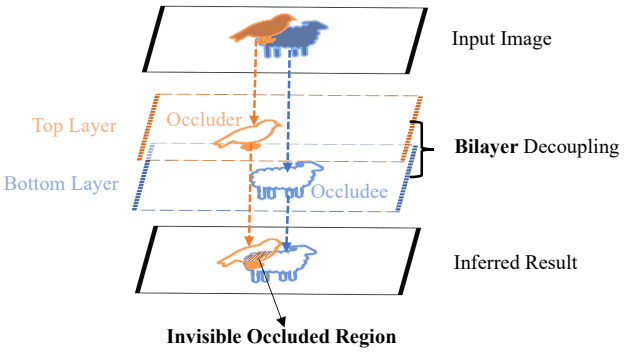

Figure 1. **Simplified illustration**. Unlike previous segmentation approaches operating on a single image layer (i.e., directly on the input image), we decouple overlapping objects into *two image layers*, where the top layer deals with the occluding objects (**occluder**) and the bottom layer for **occludee** (which is also referred to as target object in other methods as they do not explicitly consider the occluder). The overlapping parts of the two image layers indicate the invisible region of the occludee, which is explicitly modeled by our occlusion-aware BCNet framework.

We propose the Bilayer Convolutional Network (BCNet). As illustrated in Figure 1, BCNet simultaneously regresses both occluding region (occluder) and partially occluded object (occludee) after ROI extraction, which groups the pixels belonging to the occluding region and treat them equally as the pixels of the occluded object but in *two separate image layers*, and thus naturally decouples the boundaries for both objects and considers the interaction between them during the mask regression stage.

Previous approaches resolve the mask conflict between neighboring objects through non-maximum suppression or additional post-processing [43, 14, 34, 30, 20]. Consequently, their results are over-smooth along boundaries or exhibit small gaps between neighboring objects. Furthermore, since the receptive field in the ROI observes multiple objects that belong to the same class, when the occluding regions were included as part of the occluded object, traditional mask head design falls short of resolving such conflict, leaving a large portion of error as shown in Figure 2. We compare BCNet with recent amodal segmentation methods [46, 16], which predict complete ob-

[1]This research is supported in part by the Research Grant Council of the Hong Kong SAR under grant no. 16201420 and Kuaishou Technology.

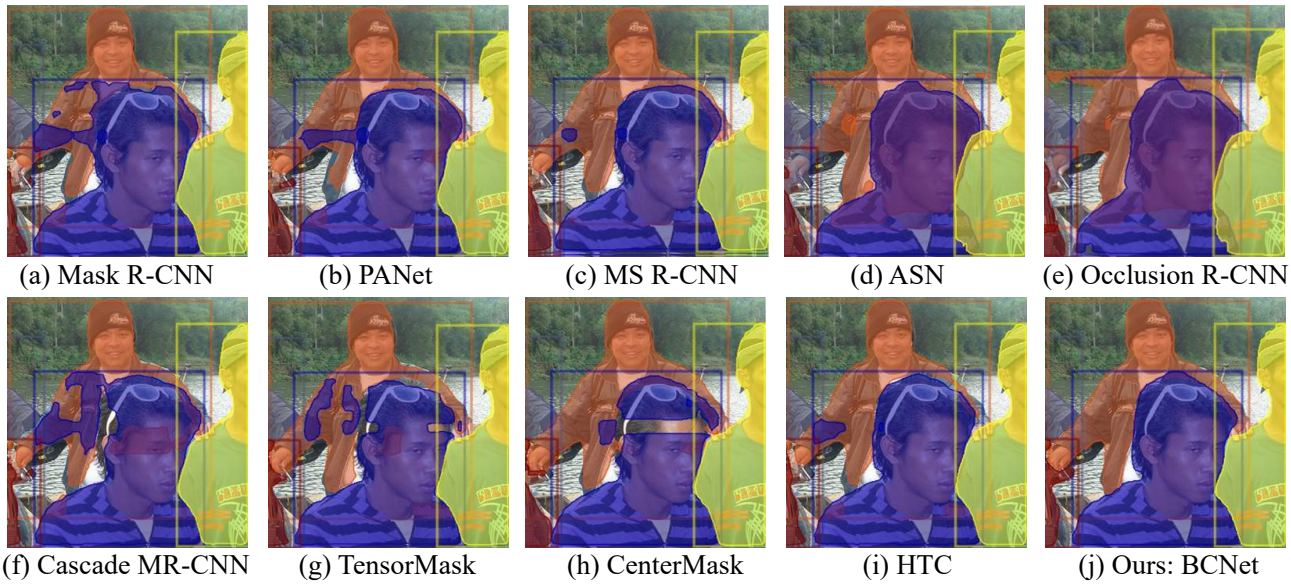

Figure 2. Instance Segmentation on **COCO** [40] validation set by a) Mask R-CNN [21], b) PANet [42], c) Mask Scoring R-CNN [25], d) ASN [46], e) Occlusion R-CNN (ORCNN) [16], f) Cascade Mask R-CNN [5], g) TensorMask [9], h) CenterMask [33], i) HTC [7] and j) Our BCNet. Note that d) and e) are specially designed for amodal/occlusion mask prediction. In this example, the bounding box is given to compare the quality of different regressed instance masks.

ject masks, including the occluded region. However, these amodal methods only regress single occluded target in the ROI, thus lacking occluder-occludee interaction reasoning, making their specially designed decoupling structure suffer when handling mask conflict between highly-overlapping objects. Correspondingly, Figure 3 compares the architecture of our BCNet with previous mask head designs [21, 42, 25, 7, 33, 5, 46, 16].

Our BCNet consists of two GCN layers with a cascaded structure, each respectively regresses the mask and boundaries of the occluding and partially occluded objects. We utilize GCN in our implementation because GCN can consider the non-local relationship between pixels, allowing for propagating information across pixels despite the presence of occluding regions. The explicit bilayer occluder-occludee relational modeling within the same ROI also makes our final segmentation results more explainable than previous methods. For object detector, we use the FCOS [51] owing to its efficient memory and running time, while noting that other state-of-the-art object detectors can also be used as demonstrated in our experiments.

Since our paper focuses on occlusion handling in instance segmentation, in addition to the original COCO evaluation, we extract a subset of COCO dataset containing both occluding objects and partially occluded objects to evaluate the robustness of our approach in comparison with other instance segmentation methods in occlusion handling. In this paper we also contribute the first large-scale occlusion aware instance segmentation datasets with ground-truth, complete object contours for *both* occluding and partially occluded objects. Extensive experiments show that our approach outperforms state-of-the-art methods in both the modal and amodal instance segmentation tasks.

2. Related Work

Instance Segmentation Two stage instance segmentation methods [37, 21, 42, 8, 5, 7, 9] achieve state-of-the-art performance by first detecting bounding boxes and then performing segmentation in each ROI region. FCIS [37] introduces the position-sensitive score maps within instance proposals for mask segmentation. Mask R-CNN [21] extends Faster R-CNN [48] with a FCN branch to segment objects in the detected box. PANet [42] further integrates multi-level feature of FPN to enhance feature representation. MS R-CNN [25] mitigates the misalignment between mask quality and score. CenterMask [33] is built upon the anchor free detector FCOS [51] with a SAG-Mask branch. In contrast, our BCNet is a *bilayer* mask prediction network for addressing the issues of heavy occlusion and overlapping objects in two-stage instance segmentation. Experiments validate that our approach leads to significant performance gain on *overall* instance segmentation performance not limited to heavily occluded cases.

One-stage instance segmentation methods remove the bounding box detection and feature re-pooling steps. AdaptIS [49] produces masks for objects located on point proposals. PolarMask [58] models instance masks in polar coordinates by instance center classification and dense distance regression. YOLOACT [4] introduces prototype masks with per-instance coefficients. SOLO [55] applies the "instance categories" concept to directly output instance masks based on the location and size. Grouping-based ap-

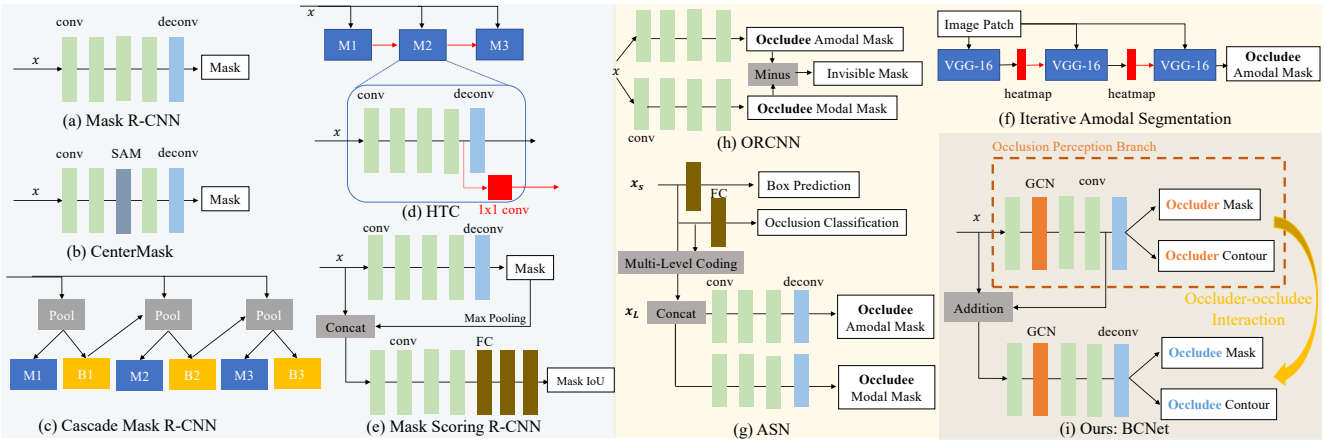

Figure 3. A brief comparison of **mask head architectures**: a) Mask R-CNN [21], b) CenterMask [33], c) Cascade Mask R-CNN [5], d) HTC [7], e) Mask Scoring R-CNN [25], f) Iterative Amodal Segmentation [35], g) ASN [46], h) ORCNN [16], where f), g) and h) are specially designed for amodal/occlusion mask prediction, i) Ours: BCNet. The input $\mathbf{x}$ denotes CNN feature after ROI extraction. `Conv` is convolution layer with 3×3 kernel, `FC` is the fully connected layer, `SAM` is the spatial attention module. B_t and M_t respectively denote box and mask head at t-th stage. Unlike previous occlusion-aware mask heads, which only regress both modal and amodal masks from the occludee, our BCNet has a *bilayer GCN structure* and considers the **interactions between the top "occluder" and bottom "occludee"** in the same ROI. The **occlusion perception branch** explicitly models the occluding object by performing joint mask and contour predictions, and distills essential occlusion information for the second graph layer to segment target object ("occludee").

proaches [28, 1, 41, 44, 3, 29] regard segmentation as a bottom-up grouping task by first producing pixel-wise predictions followed by grouping object instances in the post-processing stage. These one-stage methods, with simpler procedures than their two-stage counterparts, are more efficient but tend to be less accurate.

Occlusion Handling Methods for occlusion handling have been proposed [50, 57, 17, 10, 60, 23, 17, 65, 59]. A layout consistent random field is used in [57] to segment images of cars and faces by imposing asymmetric local spatial constraints. Ghiasi *et al*. [19] model occlusion by learning deformable models with local templates for human pose estimation while [26] reconstructs dense 3D shape for vehicle pose. Tighe *et al*. [52] build a histogram to predict occlusion overlap scores between two classes for inferring occlusion order in the scene parsing task. Chen *et al*. [12] handle occlusion by incorporating category specific reasoning and exemplar-based shape prediction for instance segmentation. For pedestrian detection with occlusion, bi-box regression is proposed in [64] for both full body and visible part estimation while repulsion loss [56] and aggregation loss [63] are designed to improve the detection accuracy. SeGAN [15] learns occlusion patterns by segmenting and generating the invisible part of an object. Recently, OCFusion [32] uses an additional branch to model instances fusion process for replacing detection confidence in panoptic segmentation. A self-supervised scene de-occlusion method is proposed in [61] by recovering the occlusion ordering and completing the mask and content for the invisible object parts.

Compared to these methods, our BCNet tackles occlusion by explicitly modeling occlusion patterns in shape and appearance. This equips the segmentation model with strong occlusion perception and reasoning capability. Our bi-layer approach can be smoothly integrated into state-of-the-art segmentation framework for end-to-end training.

Amodal Instance Segmentation Different from traditional segmentation which only focuses on visible regions, amodal instance segmentation can predict the occluded parts of object instances. Li and Malik [35] first propose a method by extending [34], which iteratively enlarges the modal bounding box following the direction of high heatmap values and synthetically adds occlusion. Zhu *et al*. [65] propose a COCO amodal dataset with 5000 images from the original COCO and use AmodalMask as a baseline, which is SharpMask [45] trained on amodal ground truth. COCOA *cls* [16] augments this dataset by assigning class-labels to the objects while SAIL-VOS dataset in [24] is targeted for video object segmentation. In autonomous driving, Qi *et al.* [46] establish the large-scale KITTI [18] InStance segmentation dataset (KINS) and present ASN to improve amodal segmentation performance.

Comparing to most of the amodal and occlusion reasoning methods which regress single occluded object boundary directly on the input (single-layered) image, our BCNet decouples overlapping objects in the same ROI into two disjoint graph layers by predicting the complete object segments (Figure 1), where the occludee is segmented under the guidance from the shape and location of the occluder.

3. Occlusion-Aware Instance Segmentation

We first give an overview to the overall instance segmentation framework, and then describe the proposed Bi-layer Graph Convolutional Network (BCNet) with explicit occluder-occludee modeling. Finally, we specify the objective functions for the whole network optimization, and provide details of training and inference process.

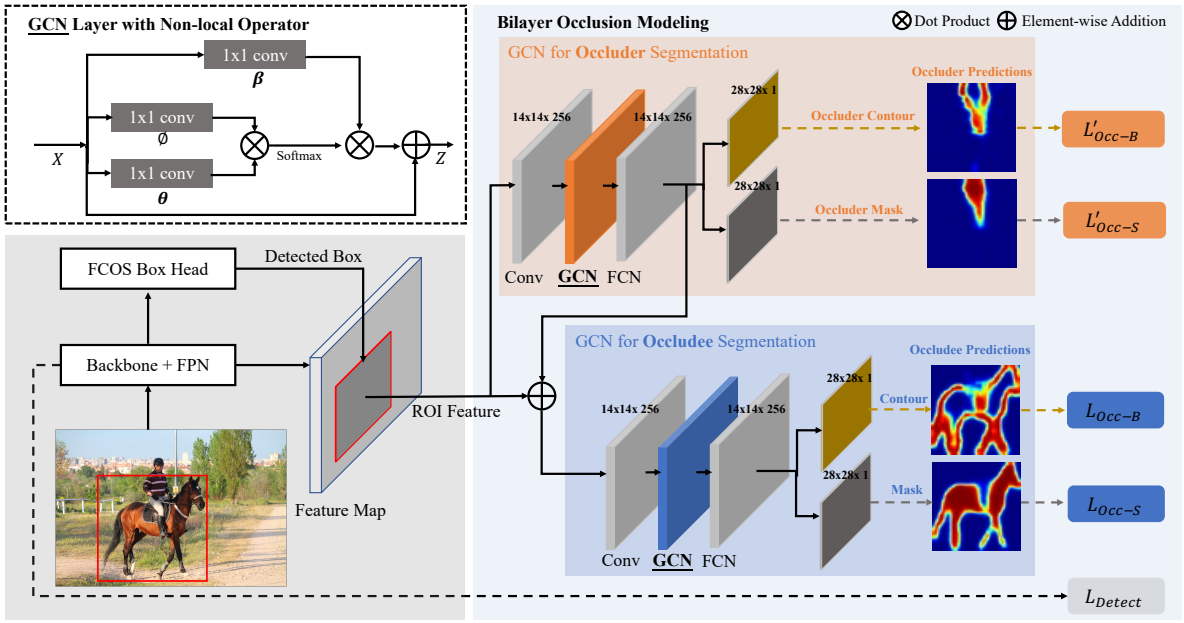

Figure 4. Architecture of our BCNet with bilayer occluder-occludee relational modeling, which consists of three modules; (1) Backbone [22] with FPN for feature extraction from input image; (2) Detection branch [51] for predicting instance proposals; (3) BCNet with bilayer GCN structure for mask prediction. For cropped ROI feature, the first GCN explicitly models occluding regions (occluder) by simultaneously detecting occlusion contours and masks, which distills essential shape and position information to guide the second GCN in mask prediction for the occludee. We utilize the non-local operator [53, 54] detailed in section 3.2 to implement the GCN layer. Visualization results are resized to square size.

3.1. Overview

Motivation For images with heavy occlusion, multiple overlapping objects in the same bounding box may result in confusing instance contours from both real objects and occlusion boundaries. The mask head design of Mask R-CNN and its variants [25, 7, 5, 46, 16] in Figure 3 directly regress the occludee with a fully convolutional network, which neglects both the occluding instances and the overlapping relations between objects. To mitigate this limitation, BCNet extends existing two stage instance segmentation methods, by adding an occlusion perception branch parallel to the traditional target prediction pipeline. Thus, the interactions between objects within the ROI region can be well considered during the mask regression stage.

Figure 4 gives the overall **architecture** of BCNet for addressing occlusion in instance segmentation. Following typical models [21, 33] for instance segmentation, our model has three parts: (1) Backbone [22] with FPN [38] for ROI feature extraction; (2) Object detection head in charge of predicting bounding boxes as instance proposals. We employ FCOS [51] as the object detector owing to its anchor-free efficiency though our method is flexible and can deploy any existing fully supervised object detectors [48, 47, 39]; (3) The *occlusion-aware mask head*, BCNet, uses bilayer GCN structure for decoupling overlapping relations and segments the instance proposals obtained from the object detection branch. BCNet reformulates the traditional class-agnostic segmentation as two complementary tasks: occluder modeling using the first GCN and occludee prediction with the second GCN, where the auxiliary predictions from the first GCN provide rich occlusion cues, such as shape and positions of occluding regions, to guide target (occludee) object segmentation.

Work Flow Given an input image, the backbone network equipped with FPN first extracts intermediate convolutional features for downstream processing. Then, the object detection head predicts bounding boxes with positions as well as categories for potential instances, and prepares the cropped ROI feature for BCNet to produce segmentation masks. The occlusion perception branch consists of the first GCN layer followed by FCN (two convolution layers), which is targeted for modeling occluding regions by jointly detecting contours and masks. Forming a residual connection, the distilled occlusion feature is element-wise added to the original input ROI feature and passed to second GCN. Finally, the second GCN, which has a similar structure to the first GCN, segments the occludee guided by this occlusion-aware feature and outputs contours and masks for the partially occluded instance.

3.2. Bilayer Occluder-Occludee Modeling

Bilayer GCN Structure for Instance Segmentation Recently, Graph Convolutional Network (GCN) [27] has been adopted to model long-range relationships in images [11, 62, 36] and videos [54]. Given highly-overlapping objects, pixels belonging to the same partially occluded object may be separated into disjoint subregions by the occluder. Thus, we adopt GCN as our basic block due to its non-local property [53], where each graph node represents a single pixel on the feature map. To explicitly model the occluding re-

gion, we further extend the single GCN block to the bilayer GCN structure as shown in Figure 4, which constructs two orthogonal graphs in a single general framework.

Following [54], given an adjacency graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with edges $\mathcal{E}$ among nodes $\mathcal{V}$, we represent the graph convolution operation as,

$$\mathbf{Z} = \sigma(\mathbf{A}\mathbf{X}\mathbf{W}_g) + \mathbf{X}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ is the input feature, $N = H \times W$ is the number of pixel grids within the ROI region and K is the feature dimension for each node, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix for defining neighboring relations of graph nodes by feature similarities, and $\mathbf{W}_g \in \mathbb{R}^{K \times K'}$ is the learnable weight matrix for the output transform, where $K' = K$ in our case. The output feature $\mathbf{Z} \in \mathbb{R}^{N \times K'}$ consists of the updated node feature by global information propagation within the whole graph layer, which is obtained after non-linear functions $\sigma(\cdot)$ including layer normalization [2] and ReLU functions. We add a residual connection after the GCN layer.

To construct the adjacency matrix $\mathbf{A}$, we define the pairwise similarity between every two graph nodes $\mathbf{x}_i, \mathbf{x}_j$ by dot product similarity as,

$$\mathbf{A}_{ij} = softmax(F(\mathbf{x}_i, \mathbf{x}_j)), \tag{2}$$

$$F(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \tag{3}$$

where θ and ϕ are two trainable transformation function implemented by 1×1 convolution as shown in the non-local operator part of Figure 4, so that high confidence edge between two nodes corresponds to larger feature similarity.

In our bilayer GCN structure, we further define $\mathcal{G}^i$ to indicate the ith graph, X_{roi} for the input ROI feature and $\mathbf{W}_f$ for weights in FCN layers, then the complete formulae are:

$$\mathbf{Z}^1 = \sigma(\mathbf{A}^1 \mathbf{X}_f \mathbf{W}_g^1) + \mathbf{X}_f, \tag{4}$$

$$\mathbf{X}_f = \mathbf{Z}^0 \mathbf{W}_f^0 + \mathbf{X}_{roi}, \tag{5}$$

$$\mathbf{Z}^0 = \sigma(\mathbf{A}^0 \mathbf{X}_{roi} \mathbf{W}_g^0) + \mathbf{X}_{roi}. \tag{6}$$

For connecting the two GCN blocks, the output feature $\mathbf{Z}^0$ of the occluder from the first GCN is directly added to $\mathbf{X}_{roi}$ to obtain the fused *occlusion-aware* feature $\mathbf{X}_f$, which is the input for the second GCN layer to output $\mathbf{Z}^1$ for occludee mask prediction.

Compared to previous class-agnostic mask head with single layer structure, where there is only binary label (foreground/background) per pixel, the bilayer GCN additionally constructs a new semantic graph space for *occluding region*. Thus a pixel node in overlapping areas in ROI can concurrently correspond to two different states in bilayer graph. While other choices may exist, we believe modeling GCN as a dual-layered structure as shown in Figure 4 is a natural choice for handling occlusion.

Occluder-occludee Modeling We explicitly model occlusion patterns by detecting both contours and masks for the occluders using the first GCN layer. Since the second GCN layer jointly predicts contours for the occludee, the overlap between the two layers can be directly identified as occlusion boundary which can thus be distinguished from real object contour (e.g., the occluder and occludee prediction on the rightmost of Figure 4). The rationale behind this design is that such irregular occlusion boundary unrelated to the occludee is confusing, which in turn provides essential cues for decoupling occlusion relations. Besides, accurate boundary localization explicitly contributes to segmentation mask prediction.

The module for occluder modeling is designed in a simple yet effective way: one 3×3 convolutional layer followed by one GCN layer and one FCN layer. Then we feed the output to the up-sampling layer and one 1×1 convolutional layer to obtain one channel feature map for joint boundary and mask predictions. The boundary detection for occluder is trained with loss $\mathcal{L}'_{\text{Occ-B}}$:

$$\mathcal{L}'_{\text{Occ-B}} = \mathcal{L}_{\text{BCE}}(W_B \mathcal{F}_{occ}(\mathbf{X}_{roi}), \mathcal{GT}_B), \tag{7}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the binary cross-entropy loss, $\mathcal{F}_{occ}$ denotes the nonlinear transformation function of the occlusion modeling module, W_B is the boundary predictor weight, $\mathbf{X}_{roi}$ is the cropped FPN feature map given by RoIAlign operation for the target region, and $\mathcal{GT}_B$ is the off-the-shelf occluder boundary that can be readily computed from mask annotations.

For occluder mask prediction, it utilizes the shared feature $\mathcal{F}_{occ}(\mathbf{X}_{roi})$, which is jointly optimized by boundary prediction. The segmentation loss $\mathcal{L}'_{\text{Occ-S}}$ for occluder modeling is designed as

$$\mathcal{L}'_{\text{Occ-S}} = \mathcal{L}_{\text{BCE}}(W_S \mathcal{F}_{occ}(\mathbf{X}_{roi}), \mathcal{GT}_S), \tag{8}$$

where W_S denotes the trainable weight of segmentation mask predictor by 1×1 convolutional layer, and $\mathcal{GT}_S$ is the mask annotations for the occluder.

3.3. End-to-end Parameter Learning

The whole instance segmentation framework can be trained in an end-to-end manner defined by a multi-task loss function $\mathcal{L}$ as,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{Detect}} + \mathcal{L}_{\text{Occluder}} + \mathcal{L}_{\text{Occludee}}, \tag{9}$$

$$\mathcal{L}_{\text{Occluder}} = \lambda_2 \mathcal{L}'_{\text{Occ-B}} + \lambda_3 \mathcal{L}'_{\text{Occ-S}} \tag{10}$$

$$\mathcal{L}_{\text{Occludee}} = \lambda_4 \mathcal{L}_{\text{Occ-B}} + \lambda_5 \mathcal{L}_{\text{Occ-S}}, \tag{11}$$

where $\mathcal{L}_{\text{Occ-B}}$ and $\mathcal{L}_{\text{Occ-S}}$ denote respectively the boundary detection and mask segmentation losses in the second GCN layer for the occludee, which are similar to Eq. 7 and Eq. 8. $\mathcal{L}_{\text{Detect}}$ supervises both the position prediction and the category classification borrowed from the FCOS [51] detector,

$$\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Regression}} + \mathcal{L}_{\text{Centerness}} + \mathcal{L}_{\text{Class}}, \tag{12}$$

and λ_1, λ_2, λ_3, λ_4 and λ_5 are hyper-parameter weights to balance the loss functions, which are tuned to be $\{1, 0.5, 0.25, 0.5, 1.0\}$ respectively on the validation set.

Training: For training the first GCN layer of BCNet, since partial occlusion cases only occupy a small fraction compared to the complete objects in COCO, we filter out part of the non-occluded ROI proposals to keep occlusion cases taking up 50% for balance sampling. SGD with momentum is employed for training 90K iterations which starts with 1K constant warm-up iterations. The batch size is set to 16 and initial learning rate is 0.01. In ablation study, ResNet-50-FPN [22] is used as backbone and the input images are resized without changing the aspect ratio by keeping the shorter side and longer side of no more than 600 and 900 pixels respectively. For leaderboard comparison, we adopt the scale-jitter where the shorter image side is randomly sampled from [640, 800] following [33, 9, 4].

Inference: During inference, the mask head predicts masks for the occluded target object in the high-score box proposals (no more than 50) generated by the FCOS detector, where the first GCN layer only produces occlusion-aware feature as input for the second GCN.

4. Experiments

4.1. Experimental Setup

COCO and COCO-OCC We conduct experiments on COCO dataset [40], where we train on 2017*train* (115k images) and evaluate results on both 2017*val* and 2017*test-dev* using the standard metrics. For further investigating segmentation performance with occlusion handling, we propose a subset split, called COCO-OCC, which contains 1,005 images extracted from the validation set (5k images) where the overlapping ratio between the bounding boxes of objects is at least 0.2. Segmenting COCO-OCC with highly overlapping objects is much more difficult than 2017*val*, where we observe a performance gap around 3.0AP for the same model in the experiment section.

KINS and COCOA We also evaluate BCNet on two amodal instance segmentation benchmarks: (1) **KINS** [46], built on the original KITTI [18], is the largest amodal segmentation benchmark for traffic scenes with both annotated amodal and modal masks for instances. BCNet is trained on the training split (7,474 images and 95,311 instances) and tested on the testing split (7,517 images and 92,492 instances) following the setting in [46]. (2) **COCOA** [65] is a subpart of COCO [40], where we train BCNet on the official training split (2,500 images) and test on the validation split (1,323 images). Note that each instance has no class label and we only use the modal and amodal mask labels for the COCOA dataset.

Synthetic Occlusion Dataset Since most objects in COCO do not exhibit significant occlusions, we synthesize a large-scale instance segmentation dataset which contains 100k images following uniform class distribution for instances among the 80 categories in COCO. Each synthetic image has *true and complete* object contours for *both* occluding and partially occluded objects, thus allowing the explicit modeling of occlusion relationship between the occlusion regions and occluded objects. On the other hand, COCOA [65], which has only 5,000 images, relies on user annotation on a given training image for "guessing" occluded object boundaries. More details on our occlusion dataset synthesis process are provided in the supplementary file.

4.2. Ablation Study

Effect of Explicit Occlusion Modeling We validate the efficacy of different components proposed for explicit occlusion modeling on the first GCN layer. Table 1 tabulates the quantitative comparison: 1) Baseline: BCNet with no explicit occlusion modeling targets; 2) modeling segmentation masks for occluding regions (**occluder**); 3) modeling contours of the occluding regions; 4) **joint** occlusion modeling on both masks and contours. Compared to the baseline, joint occlusion modeling produces the most obvious improvement especially for the heavy occlusion cases, which promotes mask AP on the standard validation set from 32.65 to 33.43, and the AP on the proposed COCO-OCC split is increased from 29.04 to 30.37.

Table 1. Effect of the first GCN for occlusion modeling by predicting contours and masks on COCO with ResNet-50-FPN model.

Occlusion (Occluder) Modeling		COCO-OCC		COCO	
Contour	Mask	AP	AP_{50}	AP	AP_{50}
		29.04	49.22	32.65	52.39
	✓	29.65	49.42	33.25	52.82
✓		30.18	49.94	33.41	53.02
✓	✓	**30.37**	**50.40**	**33.43**	**53.12**

Effect of Bilayer Occluder-occludee Modeling Built on the first GCN layer with explicit occlusion modeling, we further validate the second GCN layer in Table 2, which demonstrates the importance of *occlusion-aware* feature *guidance* for the second GCN layer to segment target object (**occludee**) by boosting 1.23 AP on COCO-OCC, and 1.06 AP on COCO respectively. Table 3 shows the results comparison on adopting the proposed *bilayer structure* and existing direct regression model with single layer. On the COCO-OCC split, bilayer GCN improves AP from 29.63 to 30.68 compared to single GCN, and bilayer FCN boosts the performance of single FCN from 28.43 to 30.12.

Table 2. Effect of the second GCN for detecting occludee contours for final mask prediction ***guided*** by the output of first GCN.

Target (Occludee) Modeling			COCO-OCC		COCO	
Guidance	Contour	Mask	AP	AP_{50}	AP	AP_{50}
		✓	29.45	49.73	32.56	52.21
✓		✓	30.37	50.40	33.43	53.12
✓	✓	✓	**30.68**	**50.62**	**33.62**	**53.26**

Using FCN or GCN? Table 3 also reveals the advantage of GCN over FCN, where GCN achieves consistent superior performance both in the singe layer and bilayer structure. We also compute the number of parameters of each model and find that although GCN has more trainable parameters, the increased model size is acceptable compared to performance gain, because the feature size of input ROI has been down-sampled to only 14×14 (spatial size) with 256 channels.

Table 3. Effect of **bilayer structure** using **GCN vs. FCN** implementation.

Structure	FCN	GCN	COCO-OCC		COCO		Params
			AP	AP_{50}	AP	AP_{50}	
Single Layer	✓		28.43	48.24	33.01	52.62	51.0M
		✓	29.63	49.59	33.14	52.81	51.4M
Bilayer	✓		30.12	49.04	33.16	52.80	53.4M
		✓	**30.68**	**50.62**	**33.62**	**53.26**	54.0M

Influence of Object Detector To investigate the influence of object detectors to BCNet, besides using one-stage detector FCOS [51], we also use representative two-stage detector Faster R-CNN [48] to perform experiments. As shown in Table 4, the performance gain brought by BCNet is consistent, with an improvement of 2.23 (for FCOS) and 2.04 (for Faster R-CNN) mask AP on COCO-OCC respectively. Here, baseline denotes mask head design in Mask R-CNN.

Table 4. Influence of the object detector (FCOS vs. Faster R-CNN) on BCNet.

Model	COCO-OCC		COCO		Params
	AP	AP_{50}	AP	AP_{50}	
FCOS [51] + Baseline	28.43	48.24	33.01	52.62	51.0M
FCOS [51] + Ours	**30.68**	**50.62**	**33.62**	**53.26**	54.0M
Faster R-CNN [48] + Baseline	29.67	49.95	33.45	53.70	60.0M
Faster R-CNN [48] + Ours	**31.71**	**51.15**	**34.61**	**54.41**	63.2M

4.3. Performance Comparison and Analysis

Comparison with SOTA Methods Table 8 compares BCNet with state-of-the-art instance segmentation methods on COCO dataset. BCGN achieves consistent improvement on different backbones and object detectors, demonstrating its effectiveness by outperforming both PANet [42] and Mask Scoring R-CNN [25] by 1.5 AP using Faster R-CNN, and exceeding CenterMask [33] by 1.3 AP using FCOS. Our single model achieves comparable result with HTC [7], which uses a 3-stage cascade refinement with multiple object detectors and mask heads, and far more parameters.

Comparison with Amodal Segmentation Methods Table 5 and Table 6 compare BCNet with other SOTA amodal segmentation methods on both the COCOA [65] and KINS [46] datasets, where: 1) AmodalMask [65] directly predicts amodal masks from image patches; 2) Occlusion RCNN (ORCNN) [16] is an extension of Mask R-CNN with both amodal and modal mask heads; 3) ASN module [46] contains additional occlusion classification branch and multi-level coding. Compared to these occlusion handling approaches, our bilayer GCN with cascaded structure still performs favorably against the state-of-the-art methods, which shows the effectiveness of BCNet in decoupling overlapping objects and mask completion under the amodal segmentation setting. Figure 5 and Figure 6 show the qualitative comparison on COCOA and KINS respectively.

Figure 5. Qualitative results comparison of the **amodal** mask predictions on **COCOA** [65] by AmodalMRCNN [16], ORCNN [16] and our method using ResNet-50, where BCNet hallucinates a more reasonable shape for the baby carriage without producing a large portion of segmentation error. We remove the "stuff" background for more clarity.

Figure 6. Qualitative results comparison of the **amodal** mask predictions on **KINS** [46] by Mask R-CNN + ASN [46] and ours, both using ResNet-101-FPN, where the boundaries of the two neighboring cars parked beside green-masked car are more reasonably estimated by BCNet.

Evaluation on Occluded Images We adopt COCO-OCC split to compare the occlusion handling ability of BCNet with other methods on images with highly overlapping objects. As shown in Table 7, our BCNet with Faster R-CNN detector has 31.71 AP *vs.* 30.32 for the Mask Scoring R-CNN [25]. By further training BCNet on the synthetic occlusion dataset, the performance of AP and AP_{50} is significantly promoted to 32.89 and 53.25 respectively, which shows the advantage brought by this new dataset.

Qualitative Evaluation. Figure 7 shows qualitative comparison of CenterMask [33] and BCNet on images with overlapping objects. In each ROI region, GCN-1 detects occluding regions while GCN-2 models the partially occluded instance by directly regressing the contours and masks. For example, BCNet decouples the occluding and occluded baseball players in similar clothes into GCN-1 and GCN-2 respectively, and detects the left leg missed by CenterMask. See supplementary file for more visual comparisons.

Table 5. Results on the COCOA dataset.

Model	AP_{all}	AP_t	AP_s
AmodalMask [65]	5.7	5.9	0.8
AmodalMRCNN [16]	21.51	21.09	9.0
ORCNN [16]	20.32	20.63	7.8
BCNet	**23.09**	**22.72**	**9.53**

Table 6. Results on the KINS dataset.

Model	AP_{Det}	AP_{Seg}
Mask R-CNN [16]	26.97	24.93
Mask R-CNN + ASN [46]	27.86	25.62
PANet [42]	27.39	25.99
PANet + ASN [46]	28.41	26.81
BCNet	**28.87**	**27.30**

Table 7. Results on COCO-OCC split.

Model	AP	AP_{50}
Mask R-CNN [22]	29.67	49.95
CenterMask [33]	29.05	49.07
MS R-CNN [25]	30.32	50.01
Ours	31.71	51.15
Ours + Synthetic	**32.89**	**53.25**

Table 8. Comparison with SOTA methods on COCO *test-dev* set. The mask AP is reported and all entries are single-model results. Note that HTC [7] adopts 3-stage cascade refinement with multiple object detectors and mask heads. All methods are trained on COCO *train2017*.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [21]	ResNet-50	35.6	57.6	38.1	18.7	38.3	46.6
PANet [42]	ResNet-50	36.6	58.0	39.3	16.3	38.1	**52.4**
BCNet + Faster R-CNN [48]	ResNet-50	**38.4**	**59.6**	**41.5**	**21.9**	**40.9**	49.3
Mask R-CNN [21]	ResNet-101	37.0	59.2	39.5	17.1	39.3	52.9
MaskLab [8]	ResNet-101	37.3	59.8	39.6	19.1	40.5	50.6
Mask Scoring R-CNN [25]	ResNet-101	38.3	58.8	41.5	17.8	40.4	54.4
BMask R-CNN [13]	ResNet-101	37.7	59.3	40.6	16.8	39.9	**54.6**
HTC [7]	ResNet-101	39.7	**61.8**	43.1	21.0	42.2	53.5
BCNet + Faster R-CNN [48]	ResNet-101	**39.8**	61.5	**43.1**	**22.7**	**42.4**	51.1
YOLACT [4]	ResNet-101	31.2	50.6	32.8	12.1	33.3	47.1
TensorMask [9]	ResNet-101	37.1	59.3	39.4	17.4	39.1	51.6
ShapeMask [31]	ResNet-101	37.4	58.1	40.0	16.1	40.1	53.8
CenterMask [33]	ResNet-101	38.3	-	-	17.7	40.8	**54.5**
BlendMask [6]	ResNet-101	38.4	60.7	41.3	18.2	41.5	53.3
BCNet + FCOS [51]	ResNet-101	**39.6**	**61.2**	**42.7**	**22.3**	**42.3**	51.0

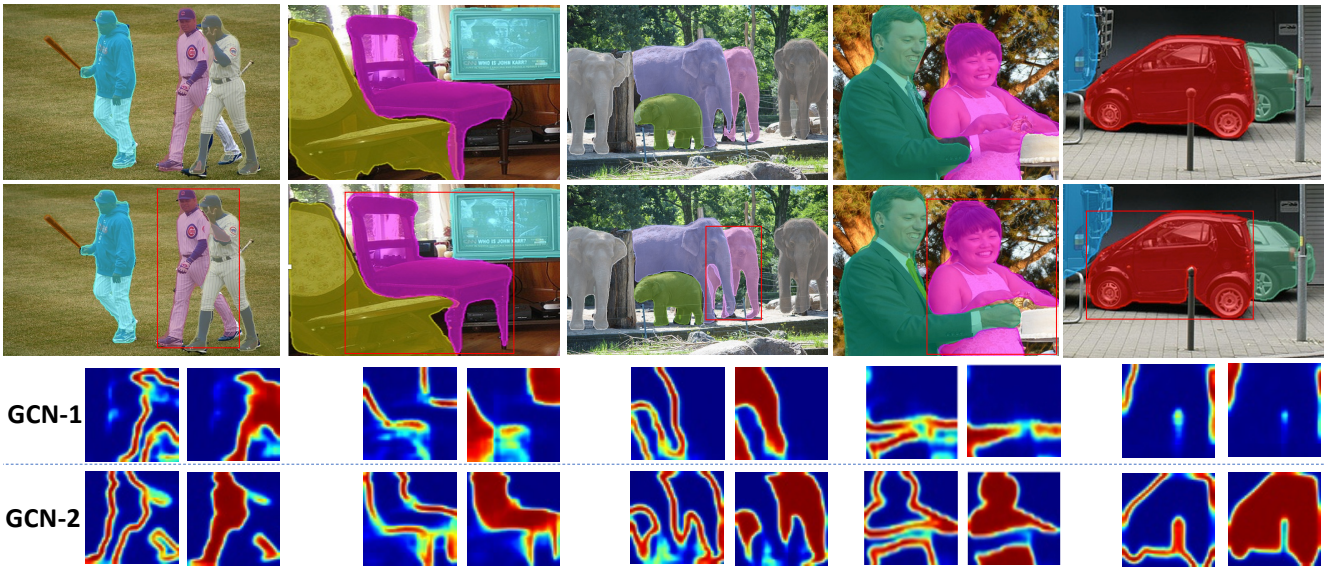

Figure 7. Qualitative instance segmentation results of CenterMask [33] (top row) and our BCNet (middle row) on **COCO** [40], both using ResNet-101-FPN and FCOS detector [51]. The bottom row visualizes squared heatmap of contour and mask predictions by the two GCN layers for the occluder and occludee in the same **ROI region** specified by the red bounding box, which also makes the final segmentation result of BCNet more explainable than previous methods. More qualitative results are available in the supplementary file.

5. Conclusion

We propose BCNet, an effective mask prediction network for addressing instance segmentation in the presence of highly-overlapping objects in two-stage instance segmentation. BCNet achieves consistent gains on overall segmentation performance using different backbones and object detectors in both the modal and amodal settings. With explicit occluder-occludee modeling, occluding and occluded instances are decoupled into two disjoint graph spaces, where the interaction between objects within each ROI region are explicitly considered. This effective approach will benefit future research in both occlusion handling and instance segmentation.

References

[1] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017. 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[3] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 3

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: real-time instance segmentation. In *ICCV*, 2019. 2, 6, 8

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1, 2, 3, 4

[6] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020. 8

[7] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 1, 2, 3, 4, 7, 8

[8] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *CVPR*, 2018. 1, 2, 8

[9] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 2, 6, 8

[10] Xianjie Chen and Alan L Yuille. Parsing occluded people by flexible compositions. In *CVPR*, 2015. 3

[11] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019. 4

[12] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015. 3

[13] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *ECCV*, 2020. 8

[14] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1

[15] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018. 3

[16] Patrick Follmann, Rebecca Kö Nig, Philipp Hä Rtinger, Michael Klostermann, and Tobias Bö Ttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *WACV*, 2019. 1, 2, 3, 4, 7, 8

[17] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011. 3

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3, 6

[19] Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C Fowlkes. Parsing occluded people. In *CVPR*, 2014. 3

[20] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 1

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 4, 8

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6, 8

[23] Edward Hsiao and Martial Hebert. Occlusion reasoning for object detectionunder arbitrary viewpoint. *PAMI*, 36(9):1803–1815, 2014. 3

[24] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *CVPR*, 2019. 3

[25] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 1, 2, 3, 4, 7, 8

[26] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *ECCV*, 2020. 3

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[28] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *CVPR*, 2017. 3

[29] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, 2018. 3

[30] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 1

[31] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, 2019. 8

[32] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *CVPR*, 2020. 3

[33] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 2, 3, 4, 6, 7, 8

[34] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016. 1, 3

[35] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016. 3

[36] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*, 2018. 4

[37] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[39] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 6, 8

[41] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 3

[42] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2, 7, 8

[43] Shu Liu, Xiaojuan Qi, Jianping Shi, Hong Zhang, and Jiaya Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *CVPR*, 2016. 1

[44] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *ECCV*, 2018. 3

[45] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016. 3

[46] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019. 1, 2, 3, 4, 6, 7, 8

[47] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 4

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 4, 7, 8

[49] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, 2019. 2

[50] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005. 3

[51] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2, 4, 5, 7, 8

[52] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 3

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4

[54] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 4, 5

[55] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019. 2

[56] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *CVPR*, 2018. 3

[57] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006. 3

[58] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020. 2

[59] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *ICCV*, 2019. 3

[60] Yi Yang, Sam Hallman, Deva Ramanan, and Charless C Fowlkes. Layered object models for image segmentation. *PAMI*, 34(9):1731–1743, 2011. 3

[61] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene deocclusion. In *CVPR*, 2020. 3

[62] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019. 4

[63] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Occlusion-aware r-cnn: detecting pedestrians in a crowd. In *ECCV*, 2018. 3

[64] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *ECCV*, 2018. 3

[65] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 3, 6, 7, 8